*Fig.1 PRIOR ART*

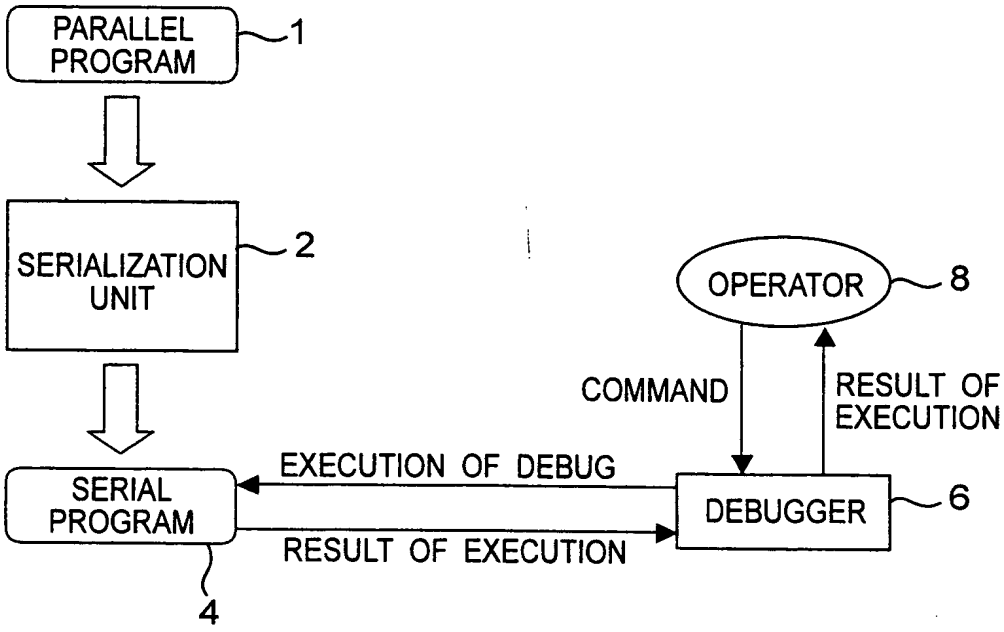


Fig.2 PRIOR ART

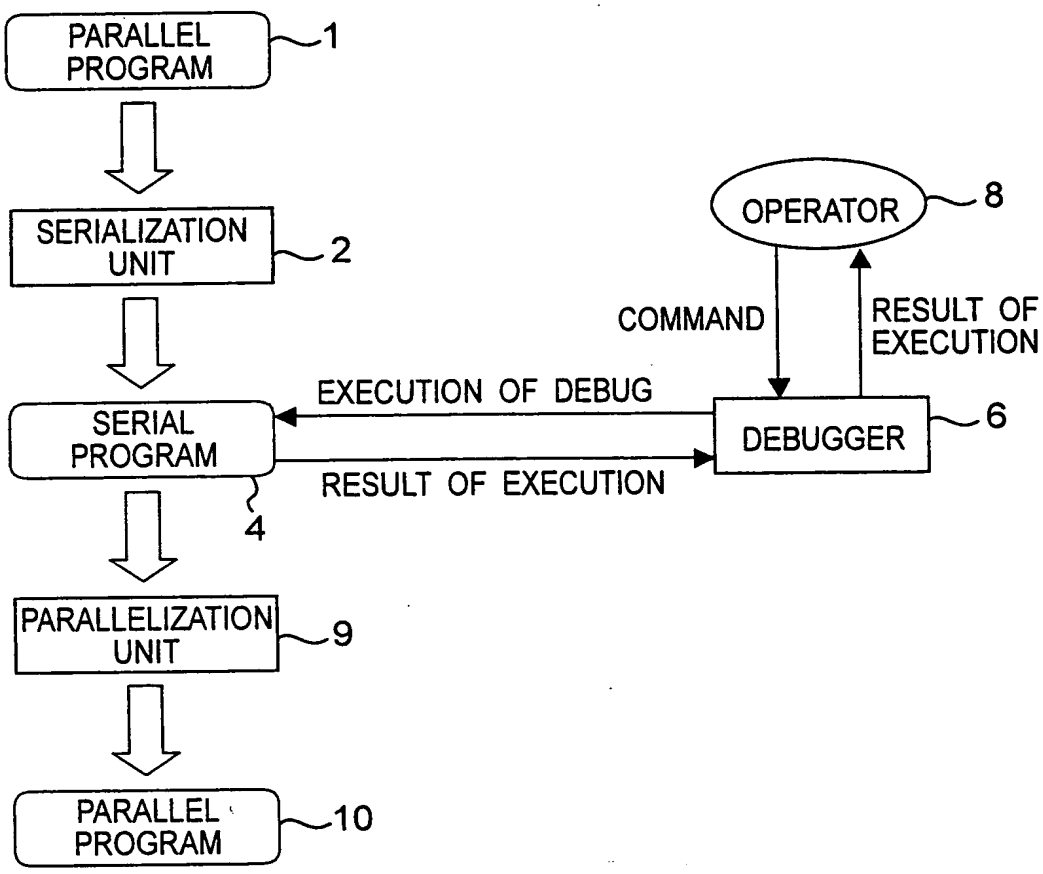


Fig.3

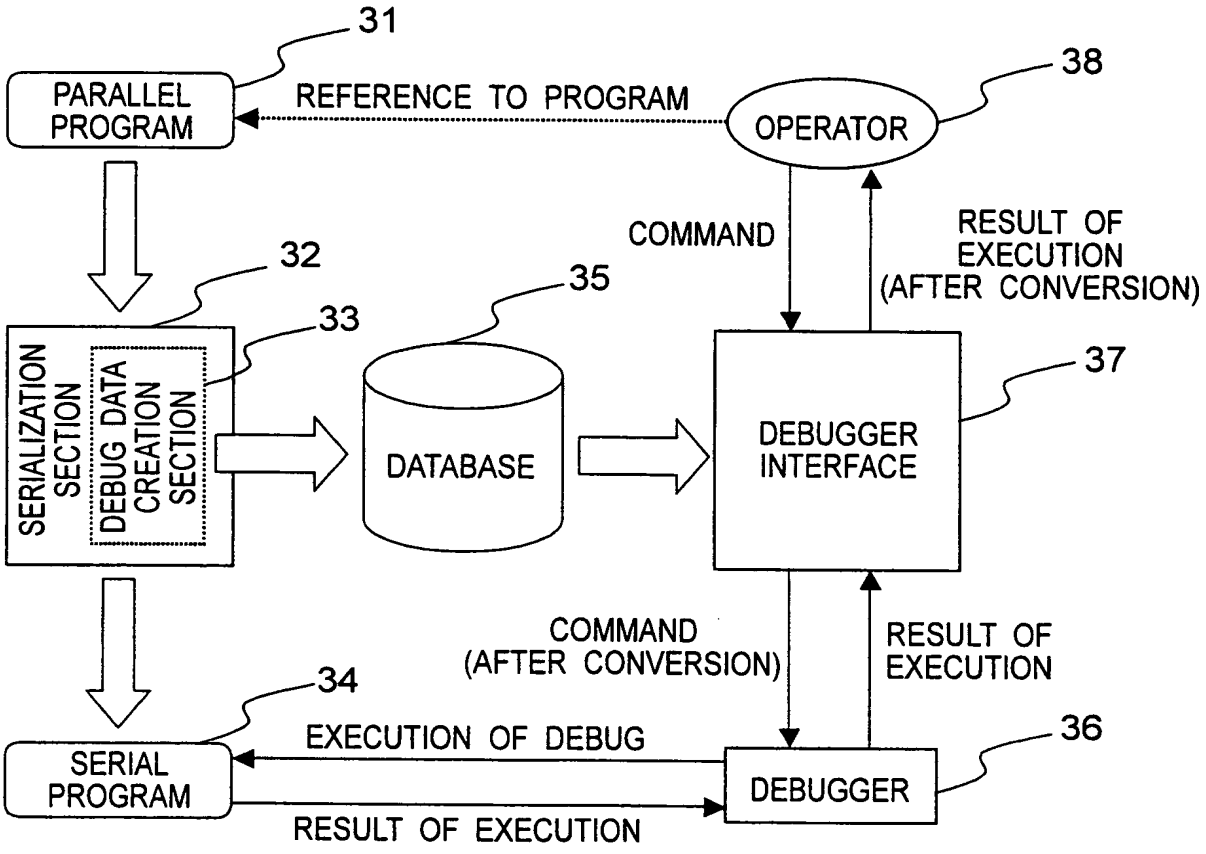

31
PARALLEL PROGRAM
REFERENCE TO PROGRAM
OPERATOR
38
COMMAND
RESULT OF EXECUTION (AFTER CONVERSION)
32
35
33
SERIALIZATION SECTION
DEBUG DATA CREATION SECTION
DATABASE
DEBUGGER INTERFACE
37
COMMAND (AFTER CONVERSION)
RESULT OF EXECUTION
34
EXECUTION OF DEBUG
SERIAL PROGRAM
DEBUGGER
36
RESULT OF EXECUTION

Fig.4

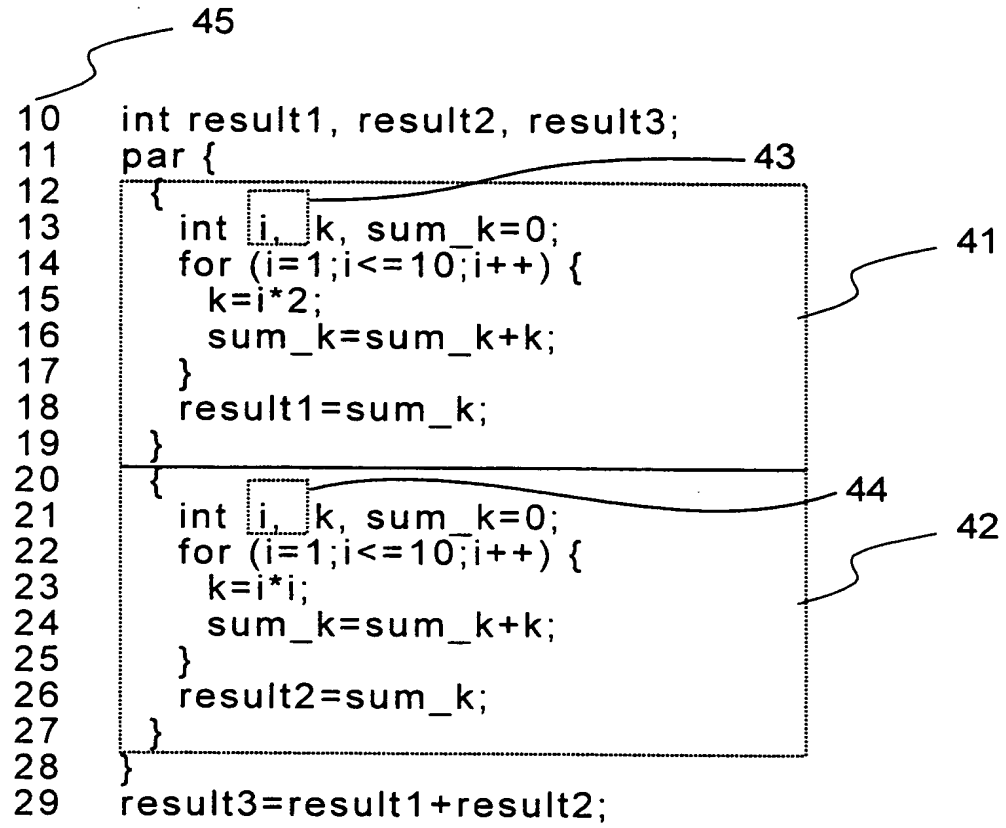

45
10 int result1, result2, result3;
11 par {
12 {
13 int i, k, sum_k=0;
14 for (i=1;i<=10;i++) {
15 k=i*2;
16 sum_k=sum_k+k;
17 }
18 result1=sum_k;
19 }
20 {
21 int i, k, sum_k=0;
22 for (i=1;i<=10;i++) {
23 k=i*i;
24 sum_k=sum_k+k;
25 }
26 result2=sum_k;
27 }
28 }
29 result3=result1+result2;
43
41
44
42

Fig.5

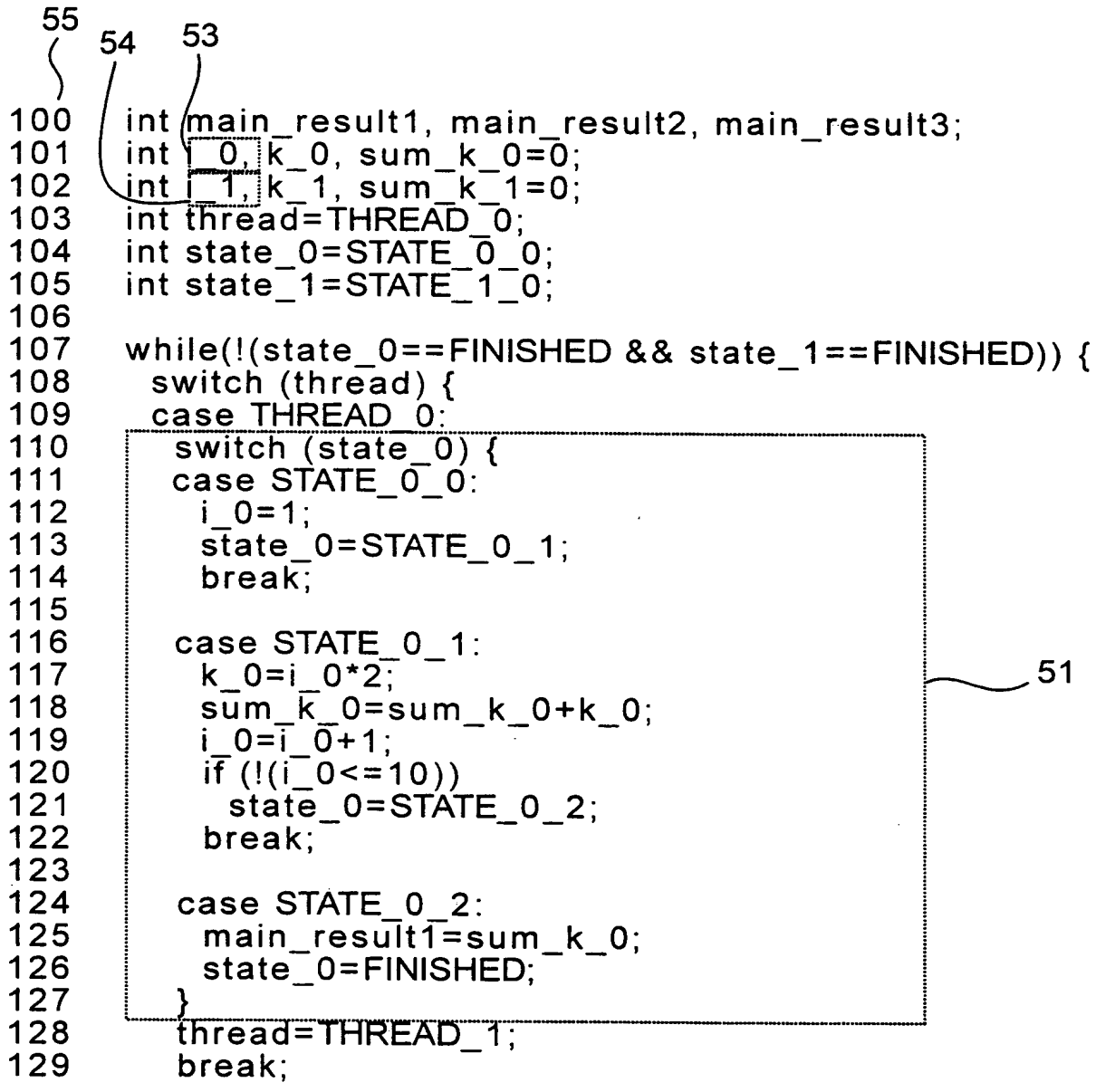

55
54
53
100 int main_result1, main_result2, main_result3;
101 int i_0, k_0, sum_k_0=0;
102 int i_1, k_1, sum_k_1=0;
103 int thread=THREAD_0;
104 int state_0=STATE_0_0;
105 int state_1=STATE_1_0;
106
107 while(!(state_0==FINISHED && state_1==FINISHED)) {
108 switch (thread) {
109 case THREAD_0:
110 switch (state_0) {
111 case STATE_0_0:
112 i_0=1;
113 state_0=STATE_0_1;
114 break;
115
116 case STATE_0_1:
117 k_0=i_0*2;
118 sum_k_0=sum_k_0+k_0;
119 i_0=i_0+1;
120 if (!(i_0<=10))
121 state_0=STATE_0_2;
122 break;
123
124 case STATE_0_2:
125 main_result1=sum_k_0;
126 state_0=FINISHED;
127 }
128 thread=THREAD_1;
129 break;
51

Fig.6

```
130
131     case THREAD_1:
132       switch (state_1) {
133       case STATE_1_0:
134         i_1=1;
135         state_1=STATE_1_1;
136         break;
137
138       case STATE_1_1:
139         k_1=i_1*i_1;
140         sum_k_1=sum_k_1+k_1;
141         i_1=i_1+1;
142         if (!(i_1<=10))
143           state_1=STATE_1_2;
144         break;
145
146       case STATE_1_2:
147         main_result2=sum_k_1;
148         state_1=FINISHED;
149       }
150       thread=THREAD_0;
151       break;
152     }
153   }
154
155 main_result3=main_result1+main_result2;
```

52

Fig.7

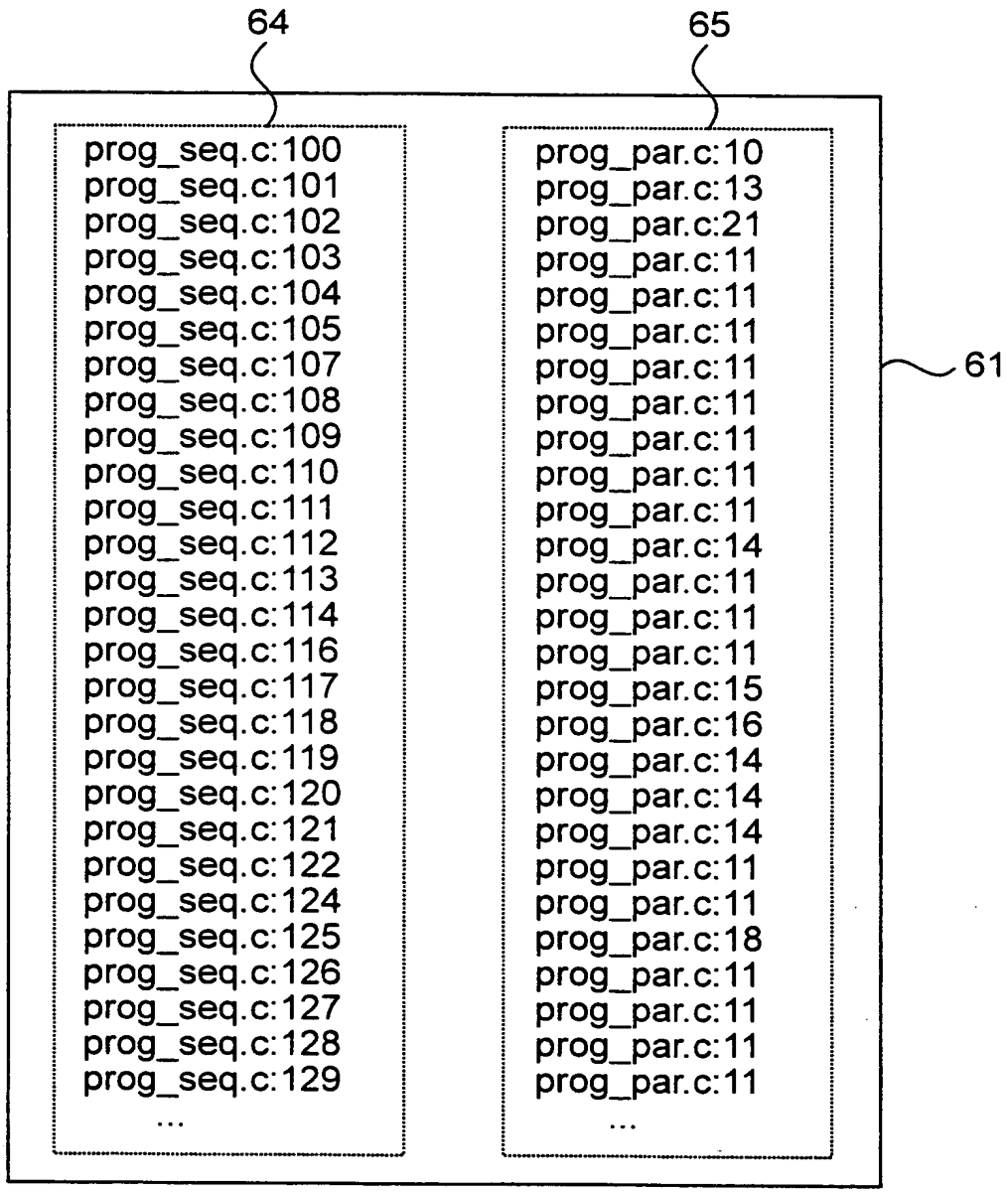

64
65
prog_seq.c:100
prog_seq.c:101
prog_seq.c:102
prog_seq.c:103
prog_seq.c:104
prog_seq.c:105
prog_seq.c:107
prog_seq.c:108
prog_seq.c:109
prog_seq.c:110
prog_seq.c:111
prog_seq.c:112
prog_seq.c:113
prog_seq.c:114
prog_seq.c:116
prog_seq.c:117
prog_seq.c:118
prog_seq.c:119
prog_seq.c:120
prog_seq.c:121
prog_seq.c:122
prog_seq.c:124
prog_seq.c:125
prog_seq.c:126
prog_seq.c:127
prog_seq.c:128
prog_seq.c:129
...
prog_par.c:10
prog_par.c:13
prog_par.c:21
prog_par.c:11
prog_par.c:11
prog_par.c:11
prog_par.c:11
prog_par.c:11
prog_par.c:11
prog_par.c:11
prog_par.c:11
prog_par.c:14
prog_par.c:11
prog_par.c:11
prog_par.c:11
prog_par.c:15
prog_par.c:16
prog_par.c:14
prog_par.c:14
prog_par.c:14
prog_par.c:11
prog_par.c:11
prog_par.c:18
prog_par.c:11
prog_par.c:11
prog_par.c:11
prog_par.c:11
...
61

Fig.8

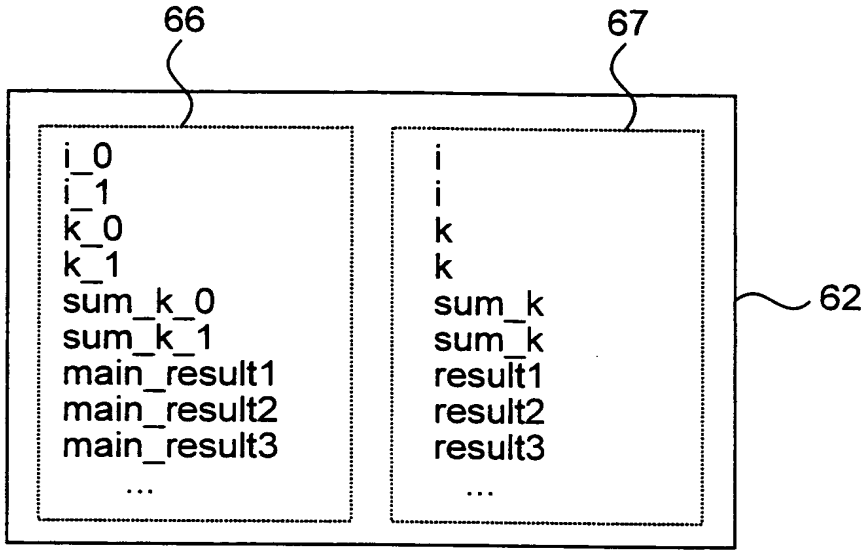

66
67
i_0
i_1
k_0
k_1
sum_k_0
sum_k_1
main_result1
main_result2
main_result3
...
i
i
k
k
sum_k
sum_k
result1
result2
result3
...
62

Fig.9

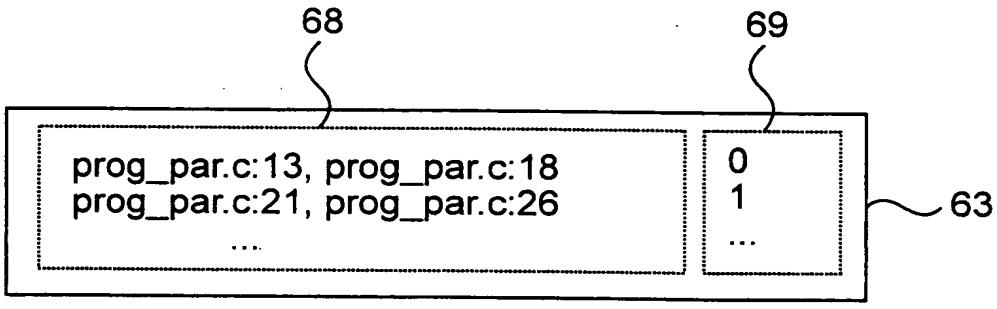

68
69
prog_par.c:13, prog_par.c:18
prog_par.c:21, prog_par.c:26
...
0
1
...
63

Fig.10

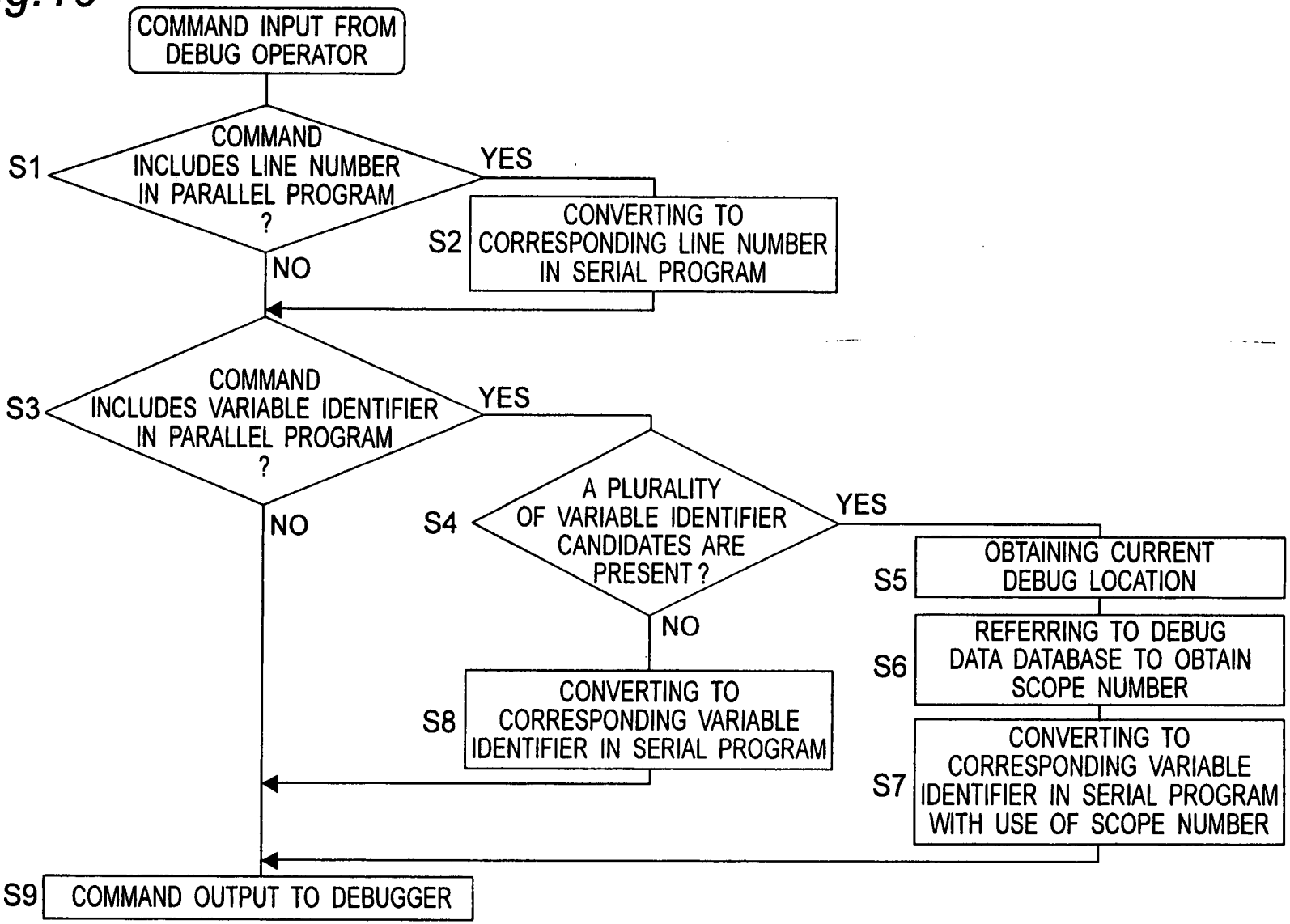
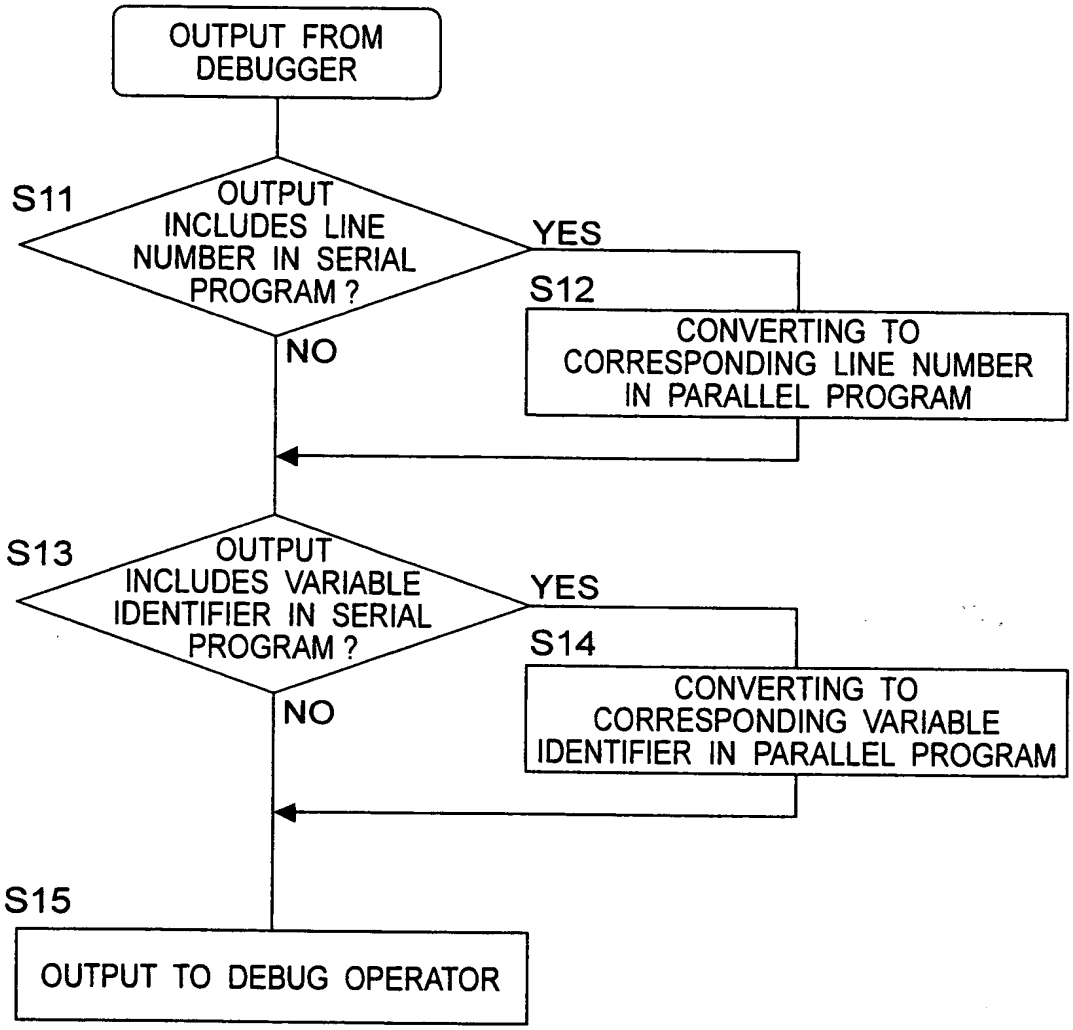

COMMAND INPUT FROM DEBUG OPERATOR
S1
COMMAND INCLUDES LINE NUMBER IN PARALLEL PROGRAM ?
YES
NO
S2
CONVERTING TO CORRESPONDING LINE NUMBER IN SERIAL PROGRAM
S3
COMMAND INCLUDES VARIABLE IDENTIFIER IN PARALLEL PROGRAM ?
YES
NO
S4
A PLURALITY OF VARIABLE IDENTIFIER CANDIDATES ARE PRESENT ?
YES
NO
S5
OBTAINING CURRENT DEBUG LOCATION
S6
REFERRING TO DEBUG DATA DATABASE TO OBTAIN SCOPE NUMBER
S7
CONVERTING TO CORRESPONDING VARIABLE IDENTIFIER IN SERIAL PROGRAM WITH USE OF SCOPE NUMBER
S8
CONVERTING TO CORRESPONDING VARIABLE IDENTIFIER IN SERIAL PROGRAM
S9
COMMAND OUTPUT TO DEBUGGER

Fig. 11


OUTPUT FROM DEBUGGER
S11
OUTPUT INCLUDES LINE NUMBER IN SERIAL PROGRAM ?
YES
S12
CONVERTING TO CORRESPONDING LINE NUMBER IN PARALLEL PROGRAM
NO
S13
OUTPUT INCLUDES VARIABLE IDENTIFIER IN SERIAL PROGRAM ?
YES
S14
CONVERTING TO CORRESPONDING VARIABLE IDENTIFIER IN PARALLEL PROGRAM
NO
S15
OUTPUT TO DEBUG OPERATOR